

Fort Worth Scene



We recently received a letter from Mr. Robert Reinert of Oley, PA. Among other items was the following information concerning our May 1979 Newsletter:

"Could you pass this information on to your software group? In the May 1979 issue (page 2) is an article on Tape Mailing List Improvements. In attempting to implement this on my Level II 16K, I ran into two problems. 1) The modifications caused a BASIC error message "OM" — Out of Memory — to occur. I got around this problem by deleting several comment lines (lines 270, 280, 2110 & 2120) which provided enough space. 2) A typographical error in your Newsletter caused the other problem; where you have

(Continued on page 2)

More Computer Clubs



Central New York TRS-80 Users Group
26 Jamesville Ave. J-4
Syracuse, New York 13210

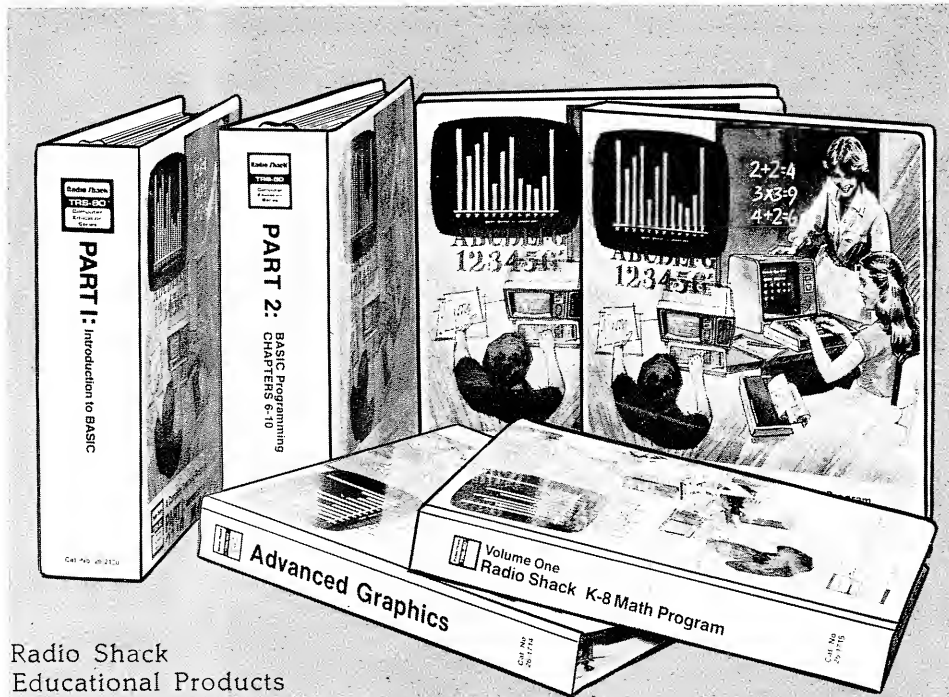
Heart of Texas Computer Club
P.O. Box 7162
Waco, Texas 76710

Madison Area TRS-80 Users
1004 E. Johnson Street
Madison, Wis. 53703
1-608/273-2180

TRS-80 Baltimore User Group
c/o Hank Becker
3505 N. Charles Street
Baltimore, Maryland 21218
1-301/358-4635

TRS-80 Users Group of Delaware
306 West 39th St.
Wilmington, Del. 19802
1-302/762-4223

COMPUTERS IN THE CLASSROOM



Radio Shack

Educational Products

shown: Euclid Geometry Tutor, Advanced Graphics, Introduction to Computer Parts 1 and 2, Essential Math Program and K-8 Math Program

The National Council of Teachers of Mathematics has adopted the following position statement:

Mathematics programs must take full advantage of the power of computers at all grade levels. Students and teachers should obtain a working knowledge of how one interacts with computers and uses their capacities.

Computer literacy is an essential outcome of contemporary education. Each student should acquire an understanding of the versatility and limitations of the computer through firsthand experiences with applications in a variety of subject matter fields. Mathematics programs designed to take full advantage of the multidisciplinary potential of computers should include the following:

- Problem Solving
- Simulations that give the opportunity to practice decision making

(Continued on page 2)

NEWSLETTER INDEX IN THIS ISSUE...

Bugs, Errors, Fixes	16
Computers in the Classroom	1
CompuServe Index	2
Educational Products Page	8
Fort Worth Scene	1
Job Jar	15
Model II	
Important Notice	3
Patch	3
Script Modification	3
More Computer Clubs	1
Product Line Manager Pages	
Color Computer	7
Model I/III	10
Model II	9
Peripherals	6
Pocket Computer	11
Redirect Model I Printer	13
View from the 7th Floor	5
What Kind of Machine?	4

Classroom (from page 1)

- Lessons that introduce and develop concepts
- Word processing to develop communication skills
- Data retrieval tasks such as career information services
- Simulations that replace dangerous, expensive, or technically difficult laboratory work
- Programming
- Drill-and-practice programs
- Student-operated computer services within the community
- Teacher-support services
- Functions that improve the evaluation process

Curriculum materials should be developed that capitalize on unique characteristics of the computer. Such materials should provide instructional experiences heretofore impossible as well as imaginative approaches to existing components of the curriculum.

The educational impact of computers is a community concern. Comprehensive planning is a joint responsibility of teachers, administrators, and parents. Each has a vital role in the definition of computer uses, the development and evaluation of software, and the selection of equipment.

According to NCTM's AN AGENDA FOR ACTION: RECOMMENDATIONS FOR SCHOOL MATHEMATICS OF THE 1980s, "public support for mathematics instruction must be raised to a level commensurate with the importance of mathematical understanding to individuals and society."

Fort Worth Scene (from page 1)

added new line 9002, there is a semi-colon shown in the middle of the line. This causes a Syntax Error. It should really be a colon, which works fine."

Thank you, Mr. Reinert, for the information. You are certainly correct about the error in line 9002. The corrected line reads:

```
9002 CLS: Z=0: P=555:
PRINT@522, "NAME OR
ZIP CODE SEQUENCE
(N/Z)": GOSUB 3000:
IF A$="Z" THEN Z=5:
ELSE IF A$<>"N"
THEN 9000
```

CompuServe™ Information Service Index

The following information is a "copy" of the CompuServe Information Service Index. We thought you might like to see the quantity and types of information which are available.

You should realize that this is only a general index, and that many of these categories have indexes of their own.

A.S.I. Monitor	Go 125
Accounting terminology	Go 13005
Adventure game	Go 49
Associated Press Wire	Go 10
Automotive Information	Go 9000
Aviation safety	Go 125
Baseball, college	Go 17
Baseball, professional	Go 17
Better Homes and Gardens	Go 1000
Blackjack game	Go 48
Bulletin Board	Go 24
Bullish or Bearish	Go 690
Business Outlook	Go 140
CB Radio Simulation	Go 39
— Instructions	Go 126
— Introduction	Go 42
Canning, home	Go 7005, 7140
Columbus Dispatch News	Go 68
Columnists:	
Dr. E on Football	Go 12005
Raylux, Financial	Go 34
Syndicated	Go 86
Commodities:	
Financial wire	Go 10
Database, how to use	Go 50
DEFAULTS, setting	Go 9
EMAIL	Go 27
— Instructions	Go 26
Editorials, local Ohio	Go 86
Electronic mail	Go 27
Energy conservation	Go 11000
Energy savers	Go 11065
Features,	
from Columbus Dispatch	Go 86
Feedback Program	Go 7
Films, reviews	Go 86, 12505
Financial advice	Go 34
Financial news:	
Columbus Dispatch	Go 85
MicroQuote	Go 29
Wire Service	Go 10
Financial Outlook	Go 142
Financial services menu	Go 21
Fireplaces, firewood	Go 11015
Food and meal planning	Go 1000
Food storage	Go 7000
Foodborne illnesses	Go 10035
Football, college	Go 17
Football, NFL:	
Injury report	Go 12300
Point spreads	Go 12105
Prognosis	Go 12005
Schedules, results	Go 12205
Wire service news	Go 67
Football computer game	Go 43

For Sale ads	Go 24
Freezing, home	Go 7070, 7205
Games, computer:	
Adventure	Go 49
Blackjack	Go 48
Football	Go 43
Spacewar	Go 44
Star Trek	Go 46
Gasoline, how to save	Go 9160
Government publications	Go 144
Health and Fitness	Go 10000
Hockey, college	Go 17
Hockey, NHL	Go 17
Home information	Go 143
Injury report, NFL	Go 12300
Investment terminology	Go 13005
Investments, MicroQuote	Go 29
Letters to the Editor	Go 86
Main Menu	Go 2
Meal plans	Go 1000
MicroQuote	Go 29
— Instructions	Go 31
Movie reviews:	
Columbus Dispatch	Go 87
CompuServe	Go 12505
NFL team summaries	Go 12130
New services	Go 800
Notices	Go 24
Nutritional Analysis	Go 1000
Ohio news	Go 82
Point spreads, NFL	Go 12105
Prescription medicines	Go 10152
Raylux, menu of services	Go 34
Recipes	Go 2000
Regional news, wire service:	
Ohio, Ind., W.Va., Penn	Go 10
Schedules, NFL teams	Go 12205
Smoking and Health	Go 10080
Space war game	Go 44
Special services	Go 33
Sports:	
Columbus Dispatch	Go 84
Wire service	Go 17
Star Trek game	Go 46
Stocks, bonds:	
Raylux advisory service	Go 34
Financial wire	Go 10
Tandy Corporation news	Go 141
Top news briefs	Go 80
U.S. Gov't Publications	Go 144
U.S. News	Go 10
U.S./World News	Go 81
User Information	Go 5
Want ads	Go 24
Washington news	Go 10
Weather:	
National, world	Go 10
Regional, (Ohio, etc.)	Go 10
What's New	Go 800
Wire service	Go 10
World news:	
Columbus Dispatch	Go 100
Wire service	Go 10

Scriptit™ Modification

Model II Scriptit gives you the ability to overstrike any character using one of 10 pre-defined symbols. It is quite possible that your printer has other symbols available if you could get to them.

A case in point is the Daisy Wheel II (26-1158). The DW II print wheel has several characters which you might wish to use, but which you can not access using Model II Scriptit.

The special overprint characters in Model II Scriptit are accessed by using the Ctrl 6, Ctrl 9, n sequence, where n is a number from 0 to 9. As you receive Model II Scriptit from the factory, the 10 special overprint characters are:

Seq	Char	Dec	Hex Value
^6^90 -	^	94	5E
^6^91 -	~	96	60
^6^92 -	~	126	7E
^6^93 -	^	167	A7
^6^94 -	..	190	BE
^6^95 -	Ç	156	9C
^6^96 -	£	163	A3
^6^97 -	μ	165	A5
^6^98 -	°	166	A6
^6^99 -	f	191	BF

Other characters which are available on the print-wheel include:

Character	Dec	Hex Value
	124	7C
à	128	80
†	168	A8
™	169	A9
®	170	AA
©	171	AB
¼	172	AC
¾	173	AD
½	174	AE
¶	175	AF
é	187	BB
û	188	BC
è	189	BD
§	192	C0
¥	204	CC
Ä	219	DB
Ö	220	DC
Ü	221	DD
ç	222	DE
ä	251	FB
ö	252	FC
ü	253	FD
ß	254	FE

To change from the defined characters to characters which you want, you will need to patch Scriptit. It is very important that if you patch Scriptit to make these changes, that you make them to a Backup copy first and make a record of what you patch! This will make future changes easier.

The ten characters which Scriptit will use are stored, beginning at location Hex BE49. On the diskette the ten locations come from the factory as:

5E607EA7BE9CA3A5A6BF

If you will compare this value with the Hex values for the ten characters, you will see that they match. The first two digits, 5E, represent the ^, the next two, 60, represent the ~, and so forth. To make a change, follow this procedure:

1. Write down the ten characters you want to use, in the order that you want them. Write them all down, even if you are only changing one value.

2. Use the two tables above to find the proper Hex values.

3. Write the ten Hex values in the same order that you have your characters in.

4. Use the following patch:

```
PATCH SCRIPSIT A=BE49
F=5E607EA7BE9CA3A5A6BF
C=XXXXXXXXXXXXXXXXXXXX
```

Except, use your ten pairs of Hex values where I wrote all of the Xs.

Here is an example—I have set my Model II Scriptit to give me the following sequence:

ç ™ ® ¢ ¼ ½ ¾ μ ° †

Looking at the tables, I find that my Hex values are:

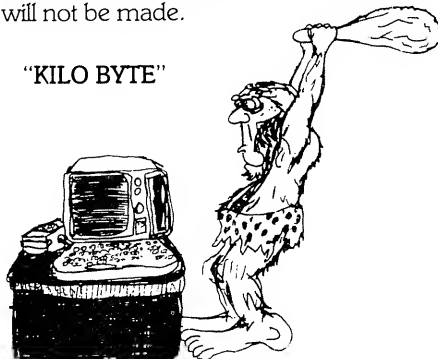
DE A9 AA AB AC AE AD A5 A6 A8

This means that my patch was:

```
PATCH SCRIPSIT A=BE49
F=5E607EA7BE9CA3A5A6BF
C=DEA9AAABACAEADA5A6A8
```

It is important that I record the value used for C, so that if I ever want to change the overstrike characters I can. If I were to make a second change, I would use the old value of C (DEA9AAABACAEADA5A6A8) for my new F = . The F stands for Find, and you are telling the computer what it will find at the address you gave it. If the computer does not find the exact sequence you give it, the patch will not be made.

"KILO BYTE"



IMPORTANT NOTICE TO MODEL II APPLICATION SOFTWARE USERS

Each Model II application software package contains a product registration card in the back of the manual.

It is very important that you fill out this card, including the product stock number (from the side of the manual), version number if indicated when you power-up the program, and your name and address.

Our records indicate that we are receiving cards for about 20% of the Model II software that we sell.

These names and addresses are being entered into Model II Profile II, and WILL be used to send information on program bugs, enhancements, and new versions. If you own a Model II program, and you do not send us the registration card, you will not get all of the information which relates to your program, including fixes or patches if errors are found at a later time.

PLEASE check all copies of your Model II software to be certain that you have sent us the registration card. If you have not sent the cards, do so now. It is for your benefit.

TRSDOS 2.0 PATCH

False Error Code 7 — Illegal Disk Change

Early reports from Model II owners indicate that some disk drives are more prone to soft read errors than others. Some media may also cause these soft read errors.

If you have a problem of getting too many false Error 7's due to these soft read errors, apply the following patches to TRSDOS:

```
PATCH SYSRES/SYS A=0E9C
F=7EB7 C=CB7E
PATCH SYSRES/SYS A=0D45
F=91 C=99
PATCH SYSRES/SYS A=0D94
F=91 C=99
```

Applying these patches will not affect system performance, but will eliminate the incorrect Error Code 7s. If you get a false Error Code 7 after these patches have been applied, your diskette contains a hard read error and must be erased and reformatted.

What Kind of Machine is it?

Last month's education article by Dennis Tanner discussed methods of writing programs which will operate in either a Model I or a Model III. Unfortunately, we had to restrict the article to Model I Level II upper-case only keyboards.

What made this unfortunate is the fact that a programmer can expect to find four distinct Level II Model I keyboard configurations (note that we are still restricting ourselves to Level II Model I's and Model III's with Model III BASIC). The first is the unmodified, upper-case only keyboard. The second variety is the Radio Shack modified, lower-case keyboard, but without the driver software. The driver software is the machine-language program which must be loaded to make the Model I produce lower-case characters on the video. The third is a keyboard with the Radio Shack lower-case modification, with the lower-case driver program in memory. The fourth variety is a keyboard with "some-one else's" lower-case or other modifications.

We can not attempt to deal with the fourth type. There are too many different modifications for us to deal with or create one universal subroutine which will determine what the configuration is.

Here is a summary of the information we need to create a program which will run in any of the Model I's we have identified, and will also run in a Model III.

1) If PEEK(293) is 73, the machine is a Model III, any other value, indicates a Model I.

2) If the machine is a Model I,

```
POKE 15360, 1;
PRINT PEEK(15360)
```

will print a 1 if the keyboard has the Radio Shack lower-case modification. If the modification has not been made, the result will be 65.

3) If the Model I keyboard has the Radio Shack lower-case modification, then

```
PRINT@0, CHR$(65);
PRINT PEEK(15360)
```

will print 65 if the lower-case driver program is installed, otherwise the result will be a 1.

4) In both the Model III, and the Model I modified for lower-case with the driver installed, the memory location 16409 is used to determine whether the keyboard is locked into upper-case only, or is in upper/lower case.

```
PRINT PEEK(16409)
```

will give a 0 if the keyboard is generating both upper and lower case. Any other value returned indicates that the keyboard is locked into upper-case only.

Here is a subroutine which you could use at the beginning of a program to determine the particular type keyboard you are in:

```
10000 TYPE=0
10010 IF PEEK(293)=73
      THEN TYPE=3:
      RETURN
10020 TEMP=PEEK(15360)
10030 POKE 15360, 1
10040 IF PEEK(15360)=65
      THEN TYPE=0: GOTO
      10070
10050 PRINT@0, CHR$(65);
10060 IF PEEK(15360)=65
      THEN TYPE=2 ELSE
      TYPE=1
10070 POKE 15360, TEMP
10080 RETURN
```

In this subroutine, TYPE is the keyboard type:

- 0—Unmodified Mod I
- 1—Modified Mod I, no driver
- 2—Modified Mod I, driver in
- 3—Model III

The variable TEMP is used to store the original value from the first video location (15360), so we can restore it when we are through.

How do we use this information? Last month, Mr. Tanner gave us the following code sequence to use for general keyboard input. The purpose of the routine is to make sure that all information input is in upper-case:

```
10 IF PEEK(293)=73 THEN
   POKE 16409,1
20 INPUT G$
30 IF G$="T" THEN PRINT
   "TRY AGAIN": GOTO 10:
   ELSE PRINT G$:GOTO 10
```

Since we now know that the Model I lower-case software uses the same memory location, we can modify line 10 of the program to read:

```
10 POKE 16409,1
```

For keyboard types 0 and 1 (Model I upper-case only and Model I lower-case without the driver), the POKE will be to an unused memory location. For keyboard types 2 and 3 (Model I with lower-case and driver, and Model III), the POKE will set the keyboard to upper-case only.

That takes care of general input, with one exception. The exception is the use of the shift key on a type 0 or 1 Model I keyboard. If the shift key is used to input any

part of the data, that part will be INPUT as lower-case. One solution to this particular problem is a scan subroutine which examines each character and converts any lower-case characters to upper-case.

A subroutine of this nature assumes that all INPUTs are done into one string variable, say A\$. Here is a sample subroutine (please note that use of this subroutine will cause a definite delay in program execution):

```
11000 C$=""
11010 FOR I%=1 TO LEN(A$)
11020 B%=MID$(A$,I%,1)
11030 B%=ASC(B$)
11040 B%=B% AND 95
11050 C$=C$+CHR$(B%)
11060 NEXT I%
11070 A$=C$
11080 RETURN
```

Line 11040 performs the "magic" of ensuring that all letters are upper-case, as well as ensuring that they are not graphics or space compression codes. By ANDing 95, we force bits 5 and 7 of each character to be a zero. Bit 5 controls upper/lower case (0 = upper, 1 = lower), and bit seven controls whether the character is alphanumeric or a graphics/space compression character (0 = alpha, 1 = graphics).

What if we need to be able to tell the difference between a shifted character and a non-shifted one? This situation arises when we use a shifted value to control some special process. Last month, Mr. Tanner gave you this routine:

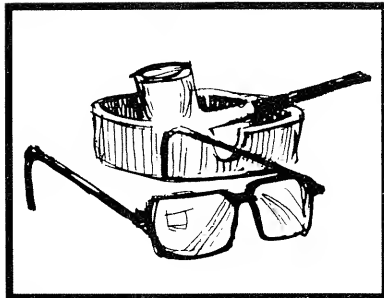
```
5 IF PEEK(293)=73 THEN
  F=32 ELSE F=0
7 IF F=32 THEN POKE
  16409,0
10 INPUT G$
20 IF ASC(G$)=116-F THEN
  PRINT "THAT'S ALL":
  END ELSE PRINT G$:
  GOTO 7
```

To make use of our new knowledge, we will modify lines 5-7 of the program to read:

```
5 GOSUB 10000
6 IF TYPE>1 THEN F=32
  ELSE F=0
7 IF F=32 THEN POKE
  16409,0
```

Below is a sample program which requests that you input your name (so we want all upper case entry) then asks you to press keys. If you press a shift Q to quit, the program ends (for this we need to distinguish between upper and lower case). Please note that for the sake of space we are not reprinting the subroutines. You do need to enter them however.

(Continued on page 16)



View From the 7th Floor

by Jon Shirley, Vice President Computer Division

Radio shack recently had a reorganization, caused by our continued rapid growth, in which Mr. John Roach became president and several other positions were changed. One effect of the reorganization is that we will move from the 7th floor but as of now I don't know if our view will improve or degrade. Stay tuned for further developments. Seriously, Computer Merchandising at Radio Shack has grown so fast that we have absolutely no room left. In fact we even have two young ladies that toil in our sample room every day. Some Ivory Tower.

As part of this reorganization I find myself with the customer services group. Those are the folks who man the hot lines and try to handle about 1,000 calls per day from those of you who manage to get through. I know some of you are not happy with the long delays to get your call answered or the fact that sometimes the answer is not too helpful. Well, we certainly want to improve customer services but the first thing I found out after visiting them is that they are totally out of space to expand and without more people we can't add more lines. So I have good news and bad news. The good news is that customer services will move to a greatly expanded area and the bad news is that it will not happen until the end of June. In the meantime, please be patient and consider yourself wearing a telephone headset 8 hours a day answering questions.

We get a lot of letters about customer services and they run about 50/50 favorable to unfavorable. The unfavorable letters always lack any detail. If you have a gripe about customer service, tell us what your problem was, what the response was, and who you spoke to. Letters that simply say customer service stinks deserve the circular file. We are constantly striving to improve customer services but we need your help. By the way I might point out that toll free hotlines are an exclusive feature of Radio Shack not available from other microcomputer companies.

There is another source for help and that's our computer centers. We have been opening them up so fast that one may have arrived near you without your knowing it. So, to help you find them, a list of computer centers with addresses and phone numbers follows this column. They are not only a good place to get help but they also stock and display our entire computer line plus a few items you won't find in any other Radio Shack store. Of course, they also give classes and many include a repair shop for computers only.

In the December column I gave a little pitch on CompuServe and our Videotex software. Since then, we have added a plus to the service and now Dow Jones Information Services are included as a part of Videotex. In addition to the free hour on CompuServe, you get another free hour on Dow Jones where you can sample financial news as reported in *The Wall Street Journal* and hot off the wire stock prices. The very good news is that the Videotex software is still only \$29.95 for Model I, II, III and Color Computer. Plus, if you have an Apple[®] owning friend we have Videotex for him also. If you have already purchased a Videotex package, check with your store about how to add Dow Jones. And if you don't know what I am talking about, visit one of our stores that is demonstrating CompuServe, it's the greatest thing to hit microcomputers since ROM's replaced paper tape.

We get a lot of letters on the subject of software pricing and the subject has recently been a hot item on the CompuServe mail service. What is a fair price for a software package? That's

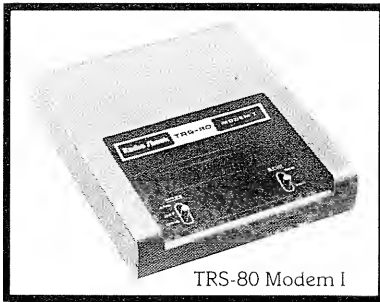
about as easy to answer as how many angels will fit on the head of a pin. First of all there is the cost to create and test the product. To create a long, complex machine language program like VisiCalc[™] or Scripsit[™] is very costly. Model II Scripsit took about 3 man years to code plus another 2 or 3 to test. That's obviously expensive, but there is another factor involved and that is potential sales. If the program has wide appeal the cost to create the program can be spread over many sales, making the software less expensive. Conversely, a narrow appeal program will have to cost more, perhaps lots more. Another hidden item is support. What do you think pays for customer service? Software pays, that's what. Our phone bill alone is \$30,000 a month and most of the calls are software related. Of course some packages are more expensive to maintain than others. A complex accounting package will generate far more phone calls than a game. What it all boils down to is that we try to set a fair, reasonable price for our software. Our word processing packages for both Model I/III and Model II cost less than the competition and do not take a back seat to anyone. We intend to keep pricing just as we have.

Finally there have been some rumors about software that you can't back up. That is partially true. Some Model III disk packages will come with a limited (2) backup capability. These disks do not have to stay in the Model III, once the program is loaded they can be removed. Why are we doing this? Obviously we do not want our software copied and distributed free. We have a right, and all our outside royalty paid writers have a right to earn a fair return on their efforts. Enough said until next month.

Here is that list of computer centers (as of 12/15/80):

Alabama	Birmingham	2428 Green Springs	205/945-0792
Arizona	Phoenix	4301 North 7th Ave	602/277-3031
	Tucson	5622 & 5626 E. Broadway	602/748-0101
Arkansas	Little Rock	Town & Country Shopping C	501/562-3202
California	Anaheim	509 Katella	714/776-9540
	Beverly Hills	8500 Wilshire Blvd.	213/659-8870
	Canoga Park	8317 Topanga Canyon	213/347-9800
	Carmichael	6305 Fair Oaks Blvd	916/484-6815
	Glendale	236 North Brand Blvd	213/246-9310
	Hayward	20942 Mission Blvd.	415/278-2888
	Long Beach	2119 Bellflower Blvd.	213/597-3377
	Riverside	Sierra Plaza	Opens Jan 81
	Sacramento	4749 J St.	916/454-3287
	San Diego	3062 Clairemont Dr.	714/276-6050
	San Francisco	One Market Plaza	415/777-9810
	San Jose	1228 South Bascom Ave.	408/297-2603
	San Mateo	3180 Campus Dr.	415/573-8607
Colorado	Colorado Sprg.	4341 N. Academy	303/593-7500
	Denver	8000 Quincy	303/770-1362
	Denver	2099 Wadsworth Blvd #M	303/232-6277
Connecticut	Manchester	228 Spencer St.	203/649-8210
	Norwalk	Rt. 7-345 Main Ave.	203/846-3418
	Orange	Woolco Shopping Center	203/795-1291
	W. Hartford	39 S. Main St.	203/523-4283
Florida	Hollywood	429 S. State Rd. #7	305/966-4382
	Miami	9459 South Dixie	305/667-2316
	Orlando	1238 E. Colonial Dr	305/894-0570
	Tampa	4555 West Kennedy	813/879-0555
	W. Palm Beach	2217 A Palm Beach Lk Blvd	305/683-1502
Georgia	Atlanta	2108 Henderson Mill	404/939-9888
	Buckhead	49 W. Paces Ferry	404/231-9604
Illinois	Buffalo Grove	1300 Dundee	312/394-0300
	Elmwood Park	7206 West Grand Ave.	312/452-7500
	Glenwood	329 Glenwood Lansing	312/758-0440
	Oaklawn	4815 W. 95th St.	312/425-9130

(Continued on page 14)



TRS-80 Modem I

Peripherals

Product Line Manager's News

I told you last month there were lots of goodies on the way from Tandy! Here begins the saga. I hardly know where to start. A place as good as any is where I left off last month.

Line Printer V

I wish I had a nickel for every phone call we received back in September when RSC-4 was published. Where there's a IV and a VI there must be a V... right? It took a while for V to get here, but we wanted it to be right and AVAILABLE! By the time you read this a good number of the new Line Printer V's (26-1165) should be showing up out there.

The supply of Line Printer III's is exhausted and the improved V (60 LPM instead of 48 for the LP III) has taken its place. No more Line Printer III's are available except those which are in stock in the stores. Line Printer V sells for \$1860 — a real bargain, Line Printer III was \$100 more. (Don't forget to buy a Printer cable to go with it.)

The NEW Modem I

If you are interested in communications, and these days who isn't, rush down to your Computer Center of R/S store to see if they have our exclusive new TRS-80 MODEM I (26-1172) on display. Our plans are to produce enough of these units so we can stock them in most stores. This new modem, that's a device designed to connect your computer to the telephone line, is a Radio Shack design, and it's a dilly. It is a direct connect modem: It connects to the phone line without using the handset. No more messed-up messages because of acoustic feed back!

The price of this compact unit is only \$149.00. More features than ever before at an even lower price. Right in the Radio Shack tradition. The unit's controls include 300/600 baud, and answer/originate switches. It is FCC approved and includes a modular jack for connection of a telephone. The modem has the standard DB-25 RS-232C connector but there is also a 4 pin DIN jack for use with the color computer.

FOR YOU MODEL ONE OWNERS, here's the best news yet! There is a switch on this unit which will convert its I/O levels to match the signals produced by the cassette port of the Model I. Add a 5 pin to 4 pin cable (26-3009 — \$4.95) and our special Cassette Comm software (26-1139 — \$9.95) and VOILA!! Instant

half-duplex dumb terminal! No expensive Expansion Interface or RS-232 board. The software responds to many standard communication codes including those for CompuServe®. Believe it when we say we will not leave MODEL I unsupported!!!

NOTES FLOATING TO THE TOP OF MY DESK

Those of you champing at the bit for new print wheels for DW II will have to have patience. The CUBIC and ITALIC wheels mentioned here in December will make it to the stores about JUNE. These things have l-o-n-g lead times! We will add others just as soon as they are available. Don't ask for "custom printwheels" — the cost would be in five figures! There are a number of large office supply firms that will attach one or more special characters to printwheels you supply them for modest fees, but we don't know how satisfactory this procedure is.

Ribbon cartridges for the LP III (and LP V) 26-1414 should be in plentiful supply now. We had to reject some faulty units, which produced a hole in the pipeline. Don't forget — a two million character ribbon will only print about 600 pages of average reports on 14" paper. The new price (\$13.95) for the LP III/V cartridge is comparable to and more in line with the price of the ribbons for LP II/IV. (26-1413 Three one million character ribbons for \$18.95, or \$6.98 a million for LP III/V and \$6.32 a million for LP II/IV.)

PRODUCT KNOWLEDGE CORNER

RADIO SHACK has been hard at work trying to develop comprehensive character set standards for our printers (This has not been one of our strong points up to now!) It is frustrating to a programmer to write a package with a certain response expected from a printer only to find that code "X" on printer "A" produces a different character on printer "B". The established ASCII standard is fuzzy around the edges. You really have to be careful when using codes for arrows, dashes, and the like.

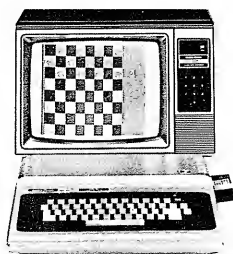
To address this problem Radio Shack has established a standard character set for its Data Processing machines. This set is resident now in Line Printers V and VI (with one exception: LP VI substitutes a left arrow for the underline for CHR\$(95)). Included are the full set of 96 ASCII American alpha-numeric characters. In addition, 24 International characters and 30 graphic symbols are also

available. These graphic characters will allow generation of all kinds of special forms and graphs. Now that these are standard we should be able to prevail upon our application programmers to use them.

This character set will be available in every printer in this class produced by Radio Shack from now on. These printers have a provision which will allow a service center to quickly adjust the response to match earlier machines to allow you to use existing software should that be necessary. This modification restores the so-called Tandy ASCII character set for codes 91 to 95 decimal. This matches the character set displayed by MODEL I. (See... SUPPORT!!!)

By next month I should be able to publish the entire standards format, including full support of data processing, word processing and graphics. (Yes I said graphics! We will have printers which can support the graphics capabilities of the Color Computer.) For now, here is a listing of some of the "special" characters which are now part of our standard printer character set:

Code			Char	Code			Char
Dec.	Hex	Oct.		Dec.	Hex	Oct.	
160	A0	240	(Blank)	224	E0	340	(Blank)
161	A1	241	à	225	E1	341	■
162	A2	242	ç	226	E2	342	■
163	A3	243	£	227	E3	343	■
164	A4	244	(Blank)	228	E4	344	■
165	A5	245	μ	229	E5	345	■
166	A6	246	°	230	E6	346	■
167	A7	247	▼	231	E7	347	■
168	A8	250	†	232	E8	350	■
169	A9	251	§	223	E9	351	■
170	AA	252	(Blank)	234	EA	352	■
171	AB	253	©	235	EB	353	■
172	AC	254	¼	236	EC	354	■
173	AD	255	(Blank)	237	ED	355	■
174	AE	256	½	238	EE	356	■
175	AF	257	¾	239	EF	357	■
176	B0	260	¥	240	F0	360	┐
177	B1	261	À	241	F1	361	—
178	B2	262	Ö	242	F2	362	┐
179	B3	263	Ü	243	F3	363	┐
180	B4	264	€	244	F4	364	┐
181	B5	265	(Blank)	245	F5	365	┐
182	B6	266	ä	246	F6	366	┐
183	B7	267	ö	247	F7	367	┐
184	B8	270	ü	248	F8	370	┐
185	B9	271	β	249	F9	371	┐
186	BA	272	(Blank)	250	FA	372	+
187	B8	273	e	251	F8	373	┐
188	8C	274	ü	252	FC	374	▲
189	BD	275	è	253	FD	375	▲
190	BE	276		254	FE	376	▲
191	BF	277	f				



Color Computer

Product Line Manager's News

As you might have noticed from the last two articles in the Newsletter, I like to get the unpleasant news out of the way early, so here goes...

PINBALL (26-3052, \$29.95)

Our PINBALL package (which I'll explain in more detail later) has been set up similar to our PERSONAL FINANCE package in one respect. When saving a board to tape, when the menu is displayed on the screen giving you instructions on how to save the data, the instant that information is displayed, the tape begins to transport if the play/record keys on the tape recorder are depressed. Just as in PERSONAL FINANCE, if you do not press "ENTER" the data will not be saved. It works the same when you are trying to load a program from tape. The thing to remember here is, if you are going to load or save a board from/to-tape, do not press the play/record buttons until you are ready to press "ENTER." This will insure that you will have a successful load/dump of your data.

CHECKERS (26-3055, \$29.95)

The other piece of news is about our CHECKERS program for the Color Computer. The "X" command discussed in the manual on page 1 is rather vague and might lead to a small misunderstanding. If a man has been moved and the eXchange key is pressed, based on the manual's explanation, you and the computer will exchange sides. This is close to the truth, but not quite right. What really happens is you and the computer trade position of your men and the number of your men but you still move from the same side of the board. Confused? Well, maybe an example will help. If the computer is poised for the kill with 10 men left and you have only 5 men left, you press the "X" key and you will be poised for the kill with 10 men and the computer will be left with 5. Your 10 men will be set up as a mirror image of how the computer's men were set up only on the bottom of the screen. Your men are still the pink ones and the computer's men are still orange, only the position and number of your men and the computer's men have changed. Clear yet? If you were to hit the "X" key again, you would be back in the same position as you originally were before you hit the "X" the first time. You are not limited in the number of times that you can exchange sides (oops, position), with the computer. Try it, it'll be a lot

clearer when you see what is happening on the screen. The other mistake (do we make mistakes?) is on page 3. The manual tells you to hold the Joysticks with the red fire button closest to you. It seems that the writer had an early version of the Joysticks, so for proper operation and response, the Joysticks should be held with the fire button away from you. Enough of the not so good, now on with the great...

LET'S PLAY...

TRS-80 PINBALL for the Color Computer: This exciting package is just what the name implies, an arcade type pinball game. The Joysticks (required) act as both the flippers and as the plunger for releasing the next ball. The right Joystick also allows you to "bump the table" in order to put more action or reaction on the ball.

FEATURES

Before we get into the fun of actually playing this game, let me point out a few of its features. First, PINBALL will allow any number of players from one to four, each getting five balls. Although all players start out with the same board setup, each player's current board situation is saved in memory until his turn.

SCORING

All four players' scores are shown on the screen so each knows what he has to do to beat the opponents. A player's turn is indicated by a dot which appears next to his score. The scoring goes as follows: there are big red circles called "circle pops" and are worth 50 points each time the ball hits one; there are small blue squares called "knock-outs" and are worth 150 points each time the ball hits one; however, when the ball hits a "knock-out" it will disappear until all of them have disappeared, at which time they all come back on again; the rest of the standard board is green and is referred to as "edge bumps," having no value upon hitting one. Not present in the standard board is an "edge pop." These are red and can be added as borders or boundaries on the board and are worth 50 points.

DESIGN YOUR OWN

Which brings me to EDITING the board. Yes, you can change the configuration of the standard board by adding more "knock outs" or "circle pops" or "edge pops," or simply erase it all and custom design your own playing field. When you are through creating the master

board, it can be saved to tape for future call-back (see note at beginning of article about saving/loading to/from tape).

CONTROLS

During play, the left Joystick fire button controls the left flipper and the right Joystick fire button controls the right flipper. Pull back on the right Joystick paddle and the next ball is loaded for release. Push forward on the paddle and the ball is released, but don't push too far or you will "tilt" the machine since the right paddle also controls "bumping the table." To bump the table, simply and quickly push forward on the right paddle and immediately pull it back to the center. Timing is important here since, if your bump coincides with the ball hitting something on the board, it will increase the rebound of the ball. However, if your bump is just before or just after the ball hits, then the momentum on the ball will be slowed down, just like the arcade machine. At any time while the ball is in play, you may press the break key and the motion of the ball stops. Press the space bar and the menu to edit the board appears. Select what you wish to correct, use the right joystick to do the correcting, press the space bar to get back to the edit menu and press "P" to resume your game at the point you left it. This is a nice feature in case you get interrupted and don't want to loose the ball in play.

BONUS PLAY

For each 10,000 points by a player, that player receives an extra, free ball. The number of balls remaining shows up in between the two flippers at the bottom of the screen. After a ball is lost "down the shoot," the next ball (if there is one) is automatically loaded for the next turn. At the end of the game, press either fire button and the menu to start over will be displayed. If you have a custom board, it will not be lost. It's just like putting another quarter in the machine.

TRY A GAME

Now, to the business at hand. Plug in the Joysticks. Plug in the PINBALL Program PakTM and turn on the computer. Turn up the volume on the TV a little bit (yes, it does make sounds when the ball goes bouncing around the board). Decide whether to go with the standard board, change the board or load one of your creations from tape and enter your choice. Decide how many people are going to

(Continued on page 12)

Education

Educational Product's News

Creating a BASIC Input Routine using INKEY\$ and the Video RAM by Dennis F. Tanner

The proficient programmer is probably aware of these limitations of the "INPUT" statement.

- The program cannot control the characters that are INPUT.
- The program cannot control the length of the string or number that is INPUT.
- The question mark always appears before the cursor.
- The **(CLEAR)** key will clear the screen if pressed while an INPUT statement is waiting for input.

This article will demonstrate an alternate way to INPUT data from the keyboard, using INKEY statements and the video RAM, which overcomes these shortcomings of the INPUT statement. The routine described is especially appropriate for educational applications in which the computer is used by young students.

The video RAM (or "video memory") is the area of the Model I or Model III TRS-80's memory which determines what is displayed on the monitor. This memory is at decimal locations 15360 to 16383. You can experiment with this memory using "POKE" statements to store a number in those locations. For example, the BASIC statement "POKE 16290,191" will make the graphics block corresponding to the number 191 appear in location 16290 near the bottom of the monitor. This method of "poking" values into the video RAM is the basis of this INPUT routine.

Before a routine in a program can place a number in the video RAM, it must know where to put it. In this routine, the initial cursor position is determined by this statement:

```
20 CP = PEEK(16416) +
    PEEK(16417) * 256
```

The variable CP will contain the current video RAM position of the cursor.

This information is used in this short INPUT program:

```
5 CLEAR 500
20 CP = PEEK(16416) +
    PEEK(16417) * 256
50 I$ = INKEY$
90 I$ = INKEY$
100 IF I$ = "" THEN G0
130 IF I$ = CHR$(13) THEN
    PRINT: PRINT X$: END
```

```
160 IF CP=16383 THEN G0
    ELSE POKE CP, ASC(I$)
170 X$ = X$ + I$: CP =
    CP + 1: GOTO 90
Line 5 CLEARs 500 bytes of string space.
Line 20 finds the initial cursor position.
Line 50 receives any keyboard stroke that
may have been accidentally pressed
before the loop begins.
Line 90 returns a one-character string
determined by an instantaneous
keyboard strobe.
Line 100 loops back to line 90 if no key
was pressed since the last strobe.
Line 130 ends the loop if (ENTER), which is
CHR$(13), is pressed. The extra
"PRINT" statement moves the
PRINT pointer to the next line, since
the POKE statements haven't
already moved it.
Line 160 POKes the ASCII code of the
character returned into the video
memory at location CP, if the loca-
tion is not 16383 (the last one on the
screen).
Line 170 adds the character to X$, moves
the cursor position variable one
position to the right, and returns to
the INKEY$ statement.
```

On the Model III, special keys like **(CLEAR)** and the arrow keys display a character when they are pressed because the program POKes the ASCII value of their keystrokes into the video RAM. After **(ENTER)** is pressed, they are PRINTed instead of POKed, so their appearance may be quite different.

We can add a blinking cursor to this routine with the following code:

```
40 CU = 32
60 CU = 168 - CU
70 POKE CP, CU
80 FOR J = 1 TO 10
*100 IF I$ = "" THEN
    NEXT J: GOTO 60
110 J = 10: NEXT J
*130 IF I$ = CHR$(13)
    THEN POKE CP, 32:
    PRINT: PRINT X$:
    END
*160 IF CP=16383 THEN G0
    ELSE POKE CP,
    ASC(I$)
*170 X$ = X$ + I$: CP =
    CP + 1: GOTO 60
```

(*Indicates lines from the first routine that were changed.)

Line 40 initializes the cursor character at 32, the code for a space.

Line 60 changes the cursor character to 136 if it is 32, or 32 if it is 136. This alternates a space with a graphics block at the cursor position, creating a blinking cursor.

Line 70 POKes the appropriate cursor character into the current cursor position of the video RAM.

Line 80 starts a loop.

Line 110 closes the loop if keyboard INPUT is received.

The backspace (left-arrow) key can be enabled with this code:

```
120 IF I$ = CHR$(8) AND
    X$ <> "" THEN 180
180 POKE CP, 32
190 CP = CP - 1
200 X$ = LEFT$(X$,
    LEN(X$)-1)
210 GOTO 60
```

Line 120 checks for code of the left-arrow key, CHR\$(8). If X\$ is not null, it goes to line 180.

Line 180 POKes a space into the current cursor position so the graphics block won't remain there.

Line 190 moves the cursor position back one space.

Line 200 effectively removes the last character of the string being built, X\$.

Line 210 returns to the blinking cursor at line 60.

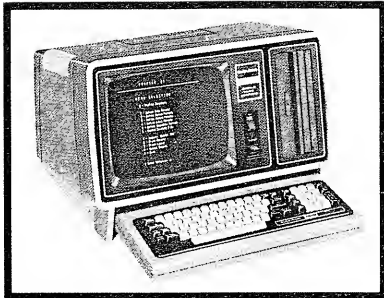
The programmer can limit the characters which will be accepted from the keyboard by adding this line:

```
140 IF I$ >= "0" AND I$
    <= "9" OR I$ = "-"
    OR I$ = "." THEN
160 ELSE G0
```

This returns to the blinking cursor if a key other than a number key, a minus sign, or a decimal point (period) is pressed. Note that the numbers 0 and 9 are in quotation marks since they are being compared to the string character stored in I\$.

The programmer probably wouldn't want to use all this code each time a INPUT from the keyboard was needed. Making this routine into a subroutine makes it much more useful. This can be accomplished, and the length of the string controlled, by adding this BASIC code:

(Continued on page 12)



Model II

Product Line Manager's News

More on TRSDOS 2.0 . . .

First . . . RADIO SHACK applications programs that were distributed on TRSDOS 1.2 should NOT be converted to 2.0. These programs were written and tested on 1.2 using only features that were available to 1.2. Even if you do convert one of the programs, the only thing you gain is the use of the alternate directory. New features, such as print spooling, are not designed into those application programs at this time.

All of the programs we release in the future will be written, tested and distributed on 2.0 and, in the near future, all of the accounting packages will be re-released with a 2.0 operating system. Registered owners of all accounting packages will be notified when these are available and how to exchange your current version for the new version. These will be available at a nominal fee to cover duplication cost and handling, but you will have to surrender the original copy of the program disk.

Owner Registration

Speaking of registered owners, all MODEL-II software packages contained a card to be returned to us with the catalog number and version number of the package you purchased. Some of the cards we have received have the catalog number of printer cables, desks, or MODEL-I software packages. If you have not mailed the card in, please do so. This is the only way we can notify you directly of enhancements or changes made to the package you own. If you don't have the card or can't find it, use a post card or letter, but get the catalog number, version, your name, and address to us.

The address for these cards is:
Computer Merchandising
700 One Tandy Center
Fort Worth, Texas 76102

Alternate Directory

Now, back to 2.0. I have had several calls regarding the speed of disk access using 2.0. It is a little slower than 1.2, but there are several reasons. One is the alternate directory. If you do a FREE from DOS, you will notice that track 48 contains "A"s. This is the alternate directory. It cannot be deleted from a disk, but you can create a backup of that disk which will not have an alternate directory. The process is simple, when you FORMAT, you use the command:

```
FORMAT :0 ALT=00
```

When you backup onto this disk, the alternate is left out. Also, if you have a disk without an alternate, you can create a backup with the alternate. During FORMAT, do not specify "ALT=00," when you do the BACKUP, the alternate directory will be created. FORMAT is the only means you have to specify that you want the alternate directory omitted.

The use of the alternate directory is completely transparent to you. There are no flags or warnings. The only way you would detect the alternate directory being used is by noticing an abnormal amount of disk drive searching during a disk access. Making a backup of that disk will re-create a good directory but only if there is an alternate to work from. The alternate directory combined with frequent backups of your disk should keep you from inadvertent, catastrophic data loss. The alternate directory is not a replacement or substitute for making backups.

Running a program without the alternate directory may gain you a little speed but at the expense of giving up your only shot in recovering from a glitched main directory.

HELP!

Have you tried the HELP command??? It has been a time saver for me. I only work with the MOD-II from 7 a.m. to 10 p.m. every day so I don't remember the proper formats and parameters for ALL of the TRSDOS commands. From DOS, try typing HELP. Then try HELP CREATE. That's the one I never seem to get right the first time. HELP, by itself, will return a list of 35 DOS commands that have further explanation available to you without having to dig out the 12 pound user manual. HELP [command] will show you the proper format, extensions and options that are available for that command. HELP CREATE, for example, returns 2 lines. Either can be used, depending on your needs at the time.

```
HELP CREATE
CREATE fs {NGRANS=nn [LRL=nn] [TYPE=<V | F>]}
CREATE fs {NRECS=nn LRL=nn [TYPE=<V | F>]}
```

fs	= filespec	Some others you may see are:	
{ }	= required	d	= drive number
[]	= optional	sw	= "ON" "OFF" c/r
< >	= choices	c/r	= ENTER key
	= one, but not both	pw	= password
nn	= number		

There are more but even these definitions are available from HELP, try typing HELP SYNTAX for the full listing. For a quick reference sheet on these, try this in TRSDOS, turn your printer on and type:

```
CLS
HELP
HELP SYNTAX
SCREEN
```

(ENTER)
(ENTER)
(ENTER)
(ENTER)

PROFILE-II users:

Have you created a one segment file, entered some data, then discovered that you needed to add another segment? There is a way to do it without ending up with garbage in the added fields.

First, make a backup of the disk "JUST IN CASE." Using the backup, from TRSDOS press **(M)** to bring up the main menu. Press **(1)** and enter the file name. Go ahead and add the field names and size to the next available segment as you normally would when creating a new segment. Press **(ENTER)** for the field name to store this information and return to the main menu. Press **(2)**, enter the file name and rework the screens as needed for the new data fields. The **(ESC)** key will store the screen map and return you to the main menu. Press **(6)** for expand, enter the file name. The system will tell you how many records you now have for that file. Write this number down then press **(BREAK)**.

When you return to the main menu, press **(X)** to go to TRSDOS READY. Rename your /KEY file to HOLD/KEY. This file name is the same as you typed in before but filled to 8 places with ZEROS followed by /KEY. I will use DJS as a file name for the example.

```
RENAME DJS000000/KEY TO HOLD/KEY
```

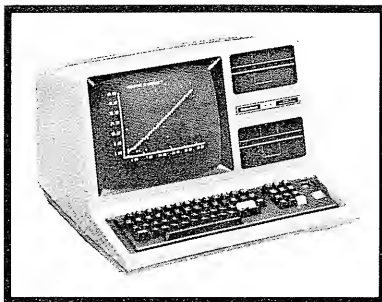
Now type:

```
CREATE DJS000000/KEY {NGRANS=0}
```

and press **(ENTER)**.

Press **(M)** and **(ENTER)** to return to the main menu. Press **(6)** to EXPAND the file. The system will tell you that you have "0" rec-

(Continued on page 12)



Model I/III

Product Line Manager's News

Let's cover some news before continuing last month's discussion of the differences between Model I and Model III.

Rumors — Limited Backups

We keep hearing the rumor that Model III owners will not be able to BACKUP TRSDOS or Model III applications software. The real situation is that Radio Shack does have the option to specify a limited number of BACKUPS but we will very carefully select those applications where it will be utilized. TRSDOS will not be protected. At this time only two products, VisiCalc[™] and Scripsit[™], are protected from an unlimited number of BACKUPS. We allow a maximum of two copies of the VisiCalc and Scripsit programs themselves. Data files, other program files and TRSDOS on these disks will BACKUP onto additional disks but the VisiCalc[™] and Scripsit[™] programs do not appear on the third and following BACKUP disks. We have included procedures with these programs for making BACKUPS, data disks, and your return of the original program disk for replacement if something goes wrong. By following these procedures you should always have a working copy to use. Since these programs don't access the disk unless you are loading or saving a data file, you can extend the life of the diskette media by loading the Scripsit[™] or VisiCalc[™] program and then removing the program disk, tucking it safely away and inserting a TRSDOS data disk. Because we are using this option only on program disks which can be loaded and then removed, legitimate users should be able to use these applications for years on the 3 available disks. Only the "pirates" should have any complaints.

TRSDOS CONVERT Utility

Model III's convert utility is designed to allow the new Model III owner to take advantage of the wealth of software already written for the Model I. Conversion is necessary since the Model III's drives utilize a double density format. Since CONVERT transfers files from your original Model I disk to a TRSDOS system disk in drive 0, it does require a two drive system. If you are CONVERTING large Model I files you may want to delete any non-system files on the Model III system disk. The Model I disks are unaltered by CONVERT. The utility does not provide for conversion of software from Model III format to Model I.

Data Conversion

In addition to the TRSDOS CONVERT utility some application programs will come with a Data Conversion program. This conversion program will allow Model I owners to take advantage of any enhancements made in the newest release of programs without having to rebuild data files. After conversion, the data can be used with this latest Model I version of the programs, or Model I owners who have purchased Model III's can then run the TRSDOS CONVERT utility to use their files on a Model III. Examples of programs with a special data conversion utility are Accounts Receivable and Accounts Payable.

Certified 40 Track Diskettes

Another item... Model III disk owners having troubles with FORMATTING and BACKUP should verify that the media you are using is suitable for Model III's 40 track double density drives. All 5¼" diskettes which we are shipping to our stores are certified for Model III use (they work fine on the Model I also). If you buy elsewhere then remember—buyer beware! You will see that we have also added a reinforcing ring to the disk to minimize any crimping of the center ring. This is the media certified for 40 track use.

Applications Programs

When a new release of TRSDOS comes out, don't be too quick to try and move all of your applications over to it. Our applications software is tested on the version of TRSDOS on which we release it. Sometimes we may use more of the available space on the new systems disk (so your programs might not fit), or we might have made some other change which could affect the operation. If you want to use a new release of TRSDOS for an existing Radio Shack application, then call customer services if you are in doubt.

Model III Differences

Now back to our discussion of some of the differences between the Model I and III. Let's continue this month with some physical differences and Model III Disk BASIC. One difference that is not obvious is the Model III's beefed up switching power supply designed to handle a full 48K of RAM, RS-232, and two internal drives. The popular Radio Shack external line filter recommended for our Model I isn't required to reduce noise and certainly isn't needed to handle Model III's single power cord. Model III's connectors

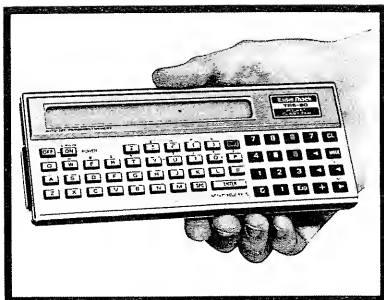
for the RS-232 (optional for the 16K Model III), I/O bus, printer jack, and disk expansion jack are all now placed neatly out of the way along the bottom rear of the Model III. Everything exits to the back of the unit. The I/O bus or expansion port has also changed from the Mod I. There is no memory control through the bus but it has been set up to allow the Model III to support interrupts from the bus. The Model III BASIC ROM allows you to specify an address for an interrupt service routine. Then, when an interrupt from a device on the bus occurs, the Model III will vector a program to the address and execute the code. The bus will be documented in the Technical Reference Manual when its available later this year.

Enhancements

We talked about the enhancements in Model III BASIC last month with the exception of the real time clock. It is now firmware supported with the TIMES function and is available to all Model III BASIC owners. We have also expanded Model III Disk BASIC to add some very powerful system commands to the Model III repertoire. An example is the command CMD"O" for sorting a string array. Specify the number of items in the memory resident string and the starting element in the array. Blearly eyed Model III owners with no other reason offered may either have been up too late with their Model III or are still trying to scan lines of their code rather than using the CMD"X" feature. It's a cross reference command which provides the line number or numbers of any string specified. It works for reserved words, variables, or strings of characters. The cross-reference will automatically output to a printer if one is available. Here is an example of both CMD"O" and CMD"X":

```
10 DEFINT N:N=6
20 DATA I,III,CC,PC,II,?
30 FOR I=1 TO N: READ A$(I):
   PRINT A$(I); " ";:NEXT I
40 PRINT
50 CMD"O", N, A$(1)
60 FOR I=1 TO N: PRINT
   A$(I); " ";:NEXT I
70 PRINT
80 CMD"X", PRINT
RUN (ENTER)
I III CC PC II ?
? CC I II III PC
00030 00040 00060
00070 00080
```

(Continued on page 12)



Pocket Computer

Product Line Manager's News

Ah... February... Spring just around the corner and Income Tax time is nearly upon us! Ugh, you say! Well, me too, but it does bring me to the subject for this month, our newly released Real Estate package for the Pocket Computer.

Now, you are probably asking what the (expletive deleted) does the Real Estate package have to do with Income Taxes. For those of you who own real estate, either your own home or income property or both, there are several modules in this package which will be of interest to you.

First, though, let's have a look at the overall package and what the various load modules are and what they will do. The package consists of 6 modules named REAL1 through REAL6.

REAL1 has two menu selections, 'Savings' and 'Loans'. The 'Savings' program will allow you to calculate any one of four unknowns (Rate in %, Number of Periods, Payment and Future Value) given the other three. The 'Loans' program provides similar functions with the addition of a balloon payment (more on this later).

REAL2 contains two programs which perform compound interest calculations, remaining balance and loan repayments.

REAL3 is a Commission Tracking program that those of you in the business can use to track and figure the commissions earned from your sales.

REAL5 consists of several programs involved in cash flow analysis. Using this module, you can calculate the present value of a sum of money to be received at some future date, or the amount to be invested today to receive a certain amount at a future date. There is also a program allowing the analysis of variable or fixed cash flows received periodically over the life of an investment and the calculation of the yield or return on invested capital, as well as testing for specific rate of return.

REAL6 is a series of five programs concerned with the various methods of figuring depreciation, i.e. Straight Line, Sum-of-the-Years Digits, Declining Balance, Excess and Composite Depreciation.

REAL 4 contains programs for the generation of a loan amortization schedule. The two programs in this module allow you to calculate an amortization

schedule payment by payment or to specify a range of payments and return the accumulated interest and principle over that period.

It is this last module that I think the majority of you will find the most useful as far as filling out your income taxes is concerned. To give you an idea of how easy this program is to use, let's take the following example:

You purchased a home during 1979 taking out a mortgage for \$55,000 at 11.75% with a principle and interest payment of \$555.18 per month and you made your first payment in July of that year. Now, you need to find out how much of the payments you made over the past year (1980) went to pay interest on your mortgage, so that you can deduct this amount from your income taxes. After you load REAL4 and select this program by pressing (SHIFT) (B), the computer will display 'LOAN RATE(%)?' you type 11.75 and press (ENTER). Next, is 'PERIODS/YR?' type 12 and (ENTER) (this tells the computer that you make monthly payments). Now you will be asked 'LOAN VALUE?' type 55000 and (ENTER). Then 'PAYMENT?' will be displayed, you type 555.18 and press (ENTER) (if you don't know what your principle and interest payment is, you could use REAL1 to find out). At this point, the computer will display 'PRE-PERIOD PAYMENT? (Y/N)', respond by typing N and pressing (ENTER). This tells the program that you make your mortgage payments in arrears, that is, the payment you make at the beginning of the month actually pays interest for the use of the mortgage money for the previous month and applies the remainder to reducing the principle amount so that the next month you will pay slightly less interest. Now, the computer will ask 'STARTING PAYMENT?', you type 7 and (ENTER) (this is because January 1980 was the seventh payment on your loan). The last question is 'NUMBER OF PAYMENTS,' type 12 and press (ENTER) (this tells the program to amortize the next 12 months payments beginning with payment number 7).

After a short delay, 'TOTAL INTEREST/PRINCIPLE' will flash briefly on the display and then you will see 'INTEREST \$5903.15.' After you have recorded this figure, press (ENTER) and the computer will display 'PRINCIPAL \$20

3,83.' As you can see, most of your payments over the year have gone for interest. If you would like to see how the ratio of interest and principle changes over the period of the loan, press (SHIFT) (B) to return to the start of the program. As each request for input comes on the screen, instead of re-typing everything simply press (ENTER). The computer will briefly flash the previous value entered and go on to the next input request. Continue pressing (ENTER) until you get to the request for 'STARTING PAYMENT?'. Now, let's say you would like to see how much interest you will pay during the fifteenth year of this mortgage, type 15*12+7 and press (ENTER). Since the computer will do the arithmetic prior to actually inputting this variable, the number 187 is entered. This represents the first month of the fifteenth calendar year. Since we want to again figure 12 payments, simply press (ENTER) after the next request. Again, after a short pause the display will show 'INTEREST \$4929.43' and pressing (ENTER) again will display 'PRINCIPAL \$1177.55'

For those of you who have investment real estate, let's look at another module in this package and an example of its use. REAL1 is the module which allows you to make various calculations concerning savings and loans; again, we will use a loan example. Assume you have a small apartment building which you agree to sell to a buyer for \$250,000. You are willing to accept a downpayment of \$35,000 and carry the balance of \$215,000 for 10 years at an interest rate of 13.5% with a balloon payment due at the end of 10 years. If the buyer makes principal and interest payments of \$2600 per month for 10 years, how much will the balloon payment be?

After the module is loaded, press (SHIFT) (B) to begin. The computer will request 'RATE(%)?' type 13.5/12 and press (ENTER). Now the program asks for '# PERIODS?', type 10*12 and (ENTER). 'PAYMENT?' is now requested so type 2600 and (ENTER). Answer N to the 'PRE-PERIOD PAYMENT?(Y/N)' question and press (ENTER). Since 'BALLOON?' is our unknown, type U and press (ENTER). The last request is 'LOAN?', you type 215000 and (ENTER). After a short pause, the computer will display 'BALLOON \$169430.36', pressing (ENTER) again

(Continued on page 12)

Color Computer (from page 7)

play and enter that choice (my but there are a lot of decisions to make in order to have fun). Finally the board appears. Let's see, there are 27 knock-outs, five circle pops, no points and four balls remaining to be played, (besides the one already in the plunger). OK, let's pick up the Joy-sticks, one in each hand with the red fire buttons pointing away from you (or leave them resting on the table, one on each side of the computer). Pull back on the right joystick paddle and on the right side of the screen a short, black, vertical line begins to appear (that's the plunger). Push forward on the paddle (up to center) and the ball is zinged out onto the playing field. Isn't it amazing how the ball just bounces around the board, hitting the knock-outs and circle pops? Just look at the points adding up! Watch out, it's heading for the flipper. Quick, press the left fire button... there, got it! Back up into the playing field to get some more points. As with any pinball machine, this one takes a little getting used to; but, with time, you will become a master and maybe even set a record (for most hours of fun on a Color Computer, maybe?).

Well, my space for this month is about used up. Happy PINBALLing and don't forget to take time out for dinner (just to keep your strength up and your reflexes sharp).

Until next month. ...

Model II (from page 9)

ords created (don't panic yet). Enter the number you wrote down previously. This will fool the system into creating that many new records for segment one (which we will kill later) and all new segments. It will also fill the data fields with blanks and the proper field terminators.

Now, when you are returned to the main menu, press (X) to go to TRSDOS ready and type:

```
KILL DJS00000/KEY
RENAME HOLD/KEY TO DJS000
00/KEY
```

(Remember to use your file name, not mine.)

You should now go into your file in the ADD/UPDATE mode from the main menu and step through your records one at a time, checking the new fields and adding data to the new fields as needed.

Model I/III (from page 10)

Calendar date conversion from any date in a year, for example, July 4, 1981 to the day of the year (in this case 185) is available with CMD "J"(julian). It also works in reverse. Here is an example:

```
10 CMD "J" , "07/04/81" ,
D$
```

```
20 PRINT D$
30 CMD "J" , "-81/200" , D$
40 PRINT D$

RUN (ENTER)
185
07/19/81
```

If you have a printer then try CMD "Z" which copies all video output to the printer. CMD "L" loads a machine language routine into RAM for access from BASIC, so if you are using a special driver or machine language routine there is no need to load it first before using your application.

Model III's enhancements to Disk BASIC include several more commands so tune in next month for those and some magic that we have added to Model III TRSDOS.

Education (from page 8)

```
10 GOTO 220
30 MX = CP + LM
*130 IF I$ = CHR$(13) AND
X$ <> "" THEN POKE
CP,32: X = VAL(X$):
PRINT: RETURN
*140 IF I$ >= "0" AND I$
<= "9" OR I$ = "-"
OR I$ = "." THEN
150 ELSE G0
150 IF CP > MX-1
THEN G0
220 CLS:PRINT "PLEASE
TYPE A NUMBER: ";
230 LM = 5
240 GOSUB 20
250 PRINT "THE NUMBER
RETURNED WAS" X ","
```

In order of execution, these lines have these functions:

Line 10 skips the subroutine and goes to the main routine.

Line 220 CLEARS the screen and displays the request for a number.

Line 230 sets the maximum string length at 5.

Line 240 goes to the subroutine beginning at line 20.

Line 30 sets the value of the maximum cursor position by adding the length to the initial position.

Line 150 RETURNS to line 60 if the string has reached its maximum length.

Line 130 RETURNS from the subroutine if the enter key, CHR\$(13), is pressed and the string X\$ is not null. Before RETURNing, it removes the cursor by poking a space into the cursor position, assigns the value of X\$ to the variable X, and goes to the next line with the "PRINT" statement.

(Note: since the numbers were being POKEd into the video RAM and not PRINTed, the "PRINT" statement is needed.)

Line 250 displays the number returned by the subroutine.

Here are some notes concerning this program:

1. Frequently used subroutines should be placed near the beginning of the program, as this one is. This speeds up the execution of the subroutine call.

2. This program could be changed to accept only capital letters by changing line 140 to this:

```
140 IF I$ < "A" OR I$ >
"Z" THEN G0
250 PRINT "THE LETTERS
RETURNED WERE ";
X$;","
```

3. The cursor character can be changed by changing the number 168 in line 60.

4. The speed of the blinking cursor can be controlled by changing the number 10 in lines 80 and 110.

5. This program can be made to work in the 32 characters/line mode by using these lines:

```
30 MX = CP + 2 * LM
160 IF CP=16382 THEN G0
ELSE POKE CP, ASC(I$)
170 X$ = X$ + I$: CP =
CP + 2: GOTO 60
190 CP = CP - 2
220 CLS: PRINT CHR$(23)
"PLEASE TYPE A
NUMBER: ";
```

6. To conserve memory space and to speed up execution, the spaces in this program could be removed and many of the lines could be combined.

Pocket Computer (from page 11)

will show 'TOTAL PMTS \$481430.36' which includes the balloon payment. Pressing (ENTER) a final time will display 'INTEREST \$266430.36'. As with the previous example, if you would like to experiment with different Interest Rates, or Payments or # of Periods etc. simply select (SHIFT) (B) and enter those variables you wish to change, type U for the unknown you wish to solve for and in short order you will have the answer.

I hope this month's column will be of some help to you in doing your annual tax chore in addition to giving you an idea of the versatility of the Real Estate package for the Pocket Computer. This package is available at your local Radio Shack Store or Computer Center under Catalog No. 26-3510 for \$24.95 and includes two program cassettes and users manual. Until March then... more Pocket Power to you.

Redirect Model I Printer Output

Here is a program for Level II Model I's which was submitted by Carey Tyler Schug of Chicago, Ill. The program will allow you to redirect line printer output to the video, and to format line printer output (whether it is going to the printer or video). The program is very fast; in fact, you probably won't even notice any delays.

Here is Carey Tyler Schug's letter (we have edited the letter for clarity):

"Here is a program (HEXPOKE2) which includes a memory loader, one which will compute relative branches as well as symbolic addresses, and can also move the contents of memory around. This latter function allows writing a program which will 'intercept' a routine already in memory, and then call the original routine.

The sample program (LPRETTY1) will intercept the LPRINT routine, insert carriage returns at the maximum width of the printer and (for all practical purposes) detect when an LLIST is being performed. If the program line is longer than the printer's line it will automatically indent successive lines of a BASIC program line (the successive lines are indented past the line number). It can also optionally start a new line every time it finds a colon or a single quote, which puts one statement per line except for 'ELSE'. Actually, the indenting and character checking is controlled by whether the printing command came from the keyboard or from executing a BASIC statement. Thus the LPRINTs in the program are not affected except by the absolute limit on the line size. The program will also automatically insert extra linefeeds to create top and bottom margins.

Because the program 'intercepts' the LPRINT routine, if an RS-232 or similar driver is loaded first, this program will function properly with printers other than those that use the parallel port.

This program will automatically adjust the top of memory pointer so that MEMORY SIZE does not need to be set. If you leave BASIC (to go to TRSDOS) and then return to BASIC, MEMORY SIZE will have to be set. (Unfortunately, no memory size was supplied.)

The data statements in the program define the machine language program as follows:

&nn	Defines a label.
*nnnnn	Defines an absolute memory location (decimal), used to poke pointers into the routine written, etc. Must come after all code which is to be relocated to top of memory.
#nn	Generates a 2-byte address of label nn.
%nn	Generates a 1-byte displacement to label nn.
-nn	Saves the current contents of memory in working storage nn (done on 1st pass only).
+nn	Pokes the contents of working storage nn.
/nn,mmmmm	Indicates the end of all data statements. nn is the number of bytes of the relocated program (checked on 1st pass to help verify CLOAD). mmmmm is a total checksum (not generated until second pass).
other	Anything else is interpreted as a two byte hexadecimal value.

Here is the program:

```

10 CLS
20 PRINT"HEXPOKE 2 /80,02,16,1520/
  Carey Tyler Schug"
30 GOSUB 1040
40 CLEAR 250 :DEFSTR A :DEFINT B-W
50 DIM S(99), L(99), T(102)
60 N=1 :T=256 :U=255 :V=16 :W=15
70 OUT 255, 4 :GOTO 90

```

```

80 M=M+N:L=L+1
90 READ A :B=ASC(A)
100 IF B>47 THEN 80
110 IF B=43 THEN 80
120 IF B=37 THEN 80
130 IF B>36 THEN 180
140 L=L+1 :M=M+N
150 IF B=35 THEN 80
160 K=LEN(A)-2 :M=M+K*N :L=L+K
170 GOTO 90
180 K=VAL(MID$(A,2))
190 IF B=38 THEN L(K)=M :GOTO 90
200 IF B=42 THEN L=K :N=0 :GOTO 90
210 IF B=45 THEN S(K)=PEEK(L) :GOTO 80
220 IF B<>47 THEN PRINT"BAD VAL" :END
230 IF M<>K THEN PRINT "BAD LEN" :END
240 IF M>128 THEN A=STRING$(128,128)
  :M=M-128 :GOTO 240
250 A= STRING$(M,M) :D=16598
260 B=VARPTR(L) :L1=PEEK(D)
270 L2=PEEK(D+1) :POKE B,L1
280 POKE B+1,L2 :L5=L
290 A=CHR$(1) :POKE 16561,PEEK(D)
300 POKE 16562,PEEK(D+1) :RESTORE
310 FOR M=1 TO 9 :T(M+48)=M :NEXT
320 FOR M=10 TO 15
330   T(M+55)=M :T(M+87)=M
340 NEXT :M=0 :N=1 :GOTO 360
350 M=M+N :L=L+1
360 READ A :B=ASC(A)
370 IF B=36 THEN 430
380 IF B<48 THEN 470
390 REM
400 C=ASC(RIGHT$(A,1))
410 D=T(B)*16+T(C) :Z=Z+D
420 POKE L,D :GOTO 350
430 FOR D=1 TO LEN(A)-1
440   E=ASC(MID$(A,D))
450   Z=Z+E :POKE L,E :L=L+1
460   NEXT :GOTO 360
470 D=VAL(MID$(A,2)) :Z=D+B*8+Z
480 IF B=38 THEN 360
490 IF B=45 THEN 350
500 IF B=42 THEN N=0 :M=0 :L=D
  :GOTO 360
510 IF B=43 THEN POKE L,S(D) :GOTO 350
520 IF B=37 THEN POKE L,L(D)-M-1
  AND U :GOTO 350
530 IF B=35 THEN C=L1 + L(D)
  :POKE L,U AND C
  :POKE L+1,C/T+L2 :L=L+2 :M=M+N+N
  :GOTO 360
540 READ Y
550 IF Y<>Z THEN PRINT "BAD SUM",Z :END
560 DATA 06, 42, &12, 00, &11, 3C
570 DATA &0, 00, &1, 00, &2, 79
580 DATA B7, CA, +0, +1, FE, 0C, 28
590 DATA 04, FE, 0D, 20, %3, 97, 47
600 DATA 32, #0, 3A, #1, E6, FE, 32
610 DATA #1, 79, FE, 0D, 28, %6, 18
620 DATA %7, &3, 3A, A3, 40, 3C, 20
630 DATA %4, 3A, #0, 47, 79, FE, 27
640 DATA 28, 04, FE, 3A, 20, 07, 3A

```

(Continued on page 14)

Printer Output (from page 13)

```

650 DATA #1, B7, 79, 28, %6, 04, FE
660 DATA 0A, 28, %6, FE, 22, 20, 09
670 DATA 3A, #1, EE, 01, 32, #1, 79
680 DATA FE, 30, 38, %5, FE, 3A, 30
690 DATA %5, 3A, #0, 3C, 21, 9B, 40
700 DATA BE, 20, %5, 32, #0, 18, %5
710 DATA &4, 06, 00, FE, 0A, 28, %6
720 DATA &5, 3A, #11, 21, 9B, 40, BE
730 DATA 30, %9, &6, C5, AF, 0E, 0D
740 DATA CD, +0, +1, C1, AF, 32, 9B
750 DATA 40, 21, #12, 7E, 3C, 77, 2B
760 DATA 96, 2B, 86, 30, %8, &7, C5
770 DATA AF, 0E, 0D, CD, +0, +1, C1
780 DATA 21, #12, 7E, 3C, 77, 2B, 96
790 DATA 38, %7, AF, 32, #12, &8, 79
800 DATA FE, 20, 78, 38, 01, 3C, 32
810 REM
820 DATA 9B, 40, 04, 18, 07, C5, 0E
830 DATA 20, CD, +0, +1, C1, 10, F7
840 DATA &9, 79, FE, 20, DB, &10
850 DATA C3, +0, +1
860 DATA *16422, -0, -1
870 DATA *16422, #2
880 DATA /196, 64387
890 INPUT "HOW MANY LINES PER PAGE";A
900 B=VAL(A)
910 IF B<6 OR B>255 THEN 890
920 INPUT "HOW MANY LINES TO SKIP AT
END OF PAGE";A
930 C=VAL(A)
940 IF C<1 OR C>B-5 THEN 920
950 INPUT "HOW MANY CHARACTERS PER
LINE"; A
960 D=VAL(A)
970 IF D<16 OR D>255 THEN 950
980 POKE LS,C :POKE LS+1,B
990 POKE LS+3,D
1000 INPUT "DO YOU WANT 1 STATEMENT
PER LINE";A
1010 IF ASC(A)=78 THEN POKE LS+5,2
:GOTO 1030
1020 IF ASC(A)=89 THEN POKE LS+5,0
1030 STOP: GOTO 1030
1040 PRINT "LPRETTY1 / 80,02,16,2145 /
Carey Tyler Schus" :PRINT
1050 I=16422
1060 IF PEEK(I) <> 141 OR
PEEK(I+1) <> 5 THEN RETURN
1070 PRINT "DO YOU WANT TO REDIRECT
THE LPRINT AND LLIST TO THE
SCREEN?";:INPUT A$
1080 IF ASC(A$)=78 THEN RETURN
1090 IF ASC(A$)=89 THEN 1110
1100 GOTO 1070
1110 POKE I,PEEK(I-8)
1120 POKE I+1,PEEK(I-7)
1130 RETURN

```

View (from page 5)

Indiana	Fort Wayne	Northcrest Shopping Cent.	219/482-9547
	Griffith	208 West Ridge Rd.	219/838-3000
	Indianapolis	6230 E. 82nd St.	317/849-6896
Iowa	Davenport	616 E. Kimberly Rd.	319/386-3457
	Des Moines	7660 Hickman Rd.	515/270-0193
Kansas	Overland Park	8619 W. 95th	913/642-1301
	Wichita	2732 Blvd. Plaza	316/681-1212

Kentucky	Louisville	2900 Taylorsville Rd.	502/459-9901
Louisiana	Baton Rouge	7007 Florida Blvd	504/928-5260
	New Orleans	327 St. Charles Ave.	Opens Jan 81
	New Orleans	3740 Veterans Hwy.	504/454-3681
Maryland	Baltimore	7942 Belair Rd.	301/882-9583
	Rockville	1633 Rockville Pike	301/984-0424
Massachusetts	Boston	200 Boylston St.	617/969-2031
	Boston	730 Commonwealth Ave.	617/739-1704
	Springfield	1985 Main St	413/732-4745
	Worcester	7 Gold Star Blvd.	Opens Feb 81
Michigan	Detroit	3620 W. Maple Rd	313/647-2151
	Detroit	1553 Woodward Ave.	313/961-6855
	Kalamazoo	25 Kalamazoo Center	517/343-0780
	Kentwood	3142 28th St. SE	Opens Feb 81
	Lansing	2519 South Cedar St.	517/372-1120
	Livonia	33470 West 7 Mile Rd.	313/476-6800
Minnesota	Bloomington	10566 France Ave	612-884-1641
	Golden Valley	Golden Valley Shopping C	612/542-8471
Missouri	Kansas City	4025 N Oak Trafficway	816/455-3381
	St. Louis	47 Florissant Oaks St.	314/921-7722
Nebraska	Omaha	3006 Dodge St.	402/346-4003
Nevada	Las Vegas	Commercial Center	702/731-3956
New Hampshire	Manchester	1000 Elm St.	603/625-4040
New Jersey	E. Brunswick	595 A Route 18	201/238-7402
	E. Hanover	Rt. 10 Hanover Plaza	201/884-1200
	Paramus	Westside Rt. 17	201/262-1920
New Mexico	Albuquerque	2108 San Mateo NE	505/265-9587
New York	Albany	Shoppers Pk, Wolf Rd	518/459-5527
	Bayshore	1751 Sunrise Hwy.	516/666-1800
	Bethpage	422 N. Wantagh Ave	516/822-6403
	New Rochelle	242 North Ave.	914/636-0700
	New York	385 Fifth Ave.	212/889-1345
	Rego Park	97-77 Queens Blvd.	212/897-5200
	Rochester	3000 Winton Rd S	716/244-6400
	Syracuse	2544 Erie Blvd.	315/446-3017
	West Seneca	4060 Seneca St	716/674-0730
North Carolina	Raleigh	6586 Glenwood Ave.	919/781-9380
	Winston-Salem	629 Peters Creek Pkwy	919/722-0030
Ohio	Canton	5248 Dressler Rd. NW	216/492-7240
	Centerville	2026 Miamisburg-	513/435-5167
	Centersv.	9725 Montgomery	513/793-8688
	Cincinnati	419 Euclid Ave	216/575-0800
	Cleveland	27561 Euclid Ave	216/261-6400
	Cleveland	838 S. Hamilton	614/864-2806
Oklahoma	Oklahoma City	4732 S.E. 29th St	405/670-4561
	Tulsa	7218 E. 41st	918/663-4488
Oregon	Eugene	390 Coburg Rd.	503/687-0177
	Portland	7463 SW Barbur Blvd	503/246-1157
Pennsylvania	Allentown	Crest Plaza SC, US 22	215/395-7155
	Harrisburg	Union Deposit Mall	717/564-6776
	Philadelphia	1002 Chestnut St.	215/923-3080
	Philadelphia	7642 Castor Ave	215/342-2217
	Pittsburgh	5775 Baptist Rd.	412/831-9694
Rhode Island	E. Providence	850 Waterman Ave	401/438-2860
South Carolina	Columbia	The Market Place	803/799-2065
	Greenville	N. Hills Shopping Center	803/292-1835
Tennessee	Memphis	4665 American Way	901/795-4963
	Nashville	2115 Franklin Pike	615/298-5484
Texas	Arlington	2500 E. Randol Mill #113	817/274-3127
	Austin	8764 E. Research Blvd.	512/459-4238
	Beaumont	5330 Easttex Freeway	713/898-7000
	Dallas	2930 W. Northwest Hwy.	214/350-4144
	Dallas	15340 Dallas Pkwy	214/934-0275
		#1100	
	Dallas	202 W. Campbell Rd.	214/669-1494
	El Paso	9515 Gateway West	915/594-8211
	Fort Worth	15 One Tandy Center	817/335-7198
	Houston	10543 Gulf Freeway	713/943-9310
	Houston	6813 Southwest Freeway	713/777-7907
	Houston	5900 North Freeway #119	713/699-1932
	San Antonio	6018 West Ave.	512/344-8792
Utah	Murray	6051 South State Ave.	801/268-8978
	Salt Lake City	415 5th Ave.	801/322-4893
Virginia	Alexandria	4527 Duke St. West End	703/370-1999
		S.C.	
	Norfolk	5731 Poplar Hall Dr.	804/428-2140
	Richmond	1617 Willow Lawn Dr.	804/282-3453
Washington	Seattle	18405 Aurora Ave. N.	206/542-6154
	Tukwila	53rd Ave. South	206/248-3710
West Virginia	Huntington	2701 1/2 Fifth Ave.	Opens Jan 81
Wisconsin	Milwaukee	6450 North 76th St.	414/353-6790
	West Allis	2727 South 108th St.	414/327-4240

Job Jar

The following program was submitted by John F. Strazzarino of South San Francisco, CA. Here are selected portions of his letter (Edited to fit available space):

"A "Job Jar" contains slips of paper with tasks to be done by the husband. Instead of having a jar sitting forever on the kitchen table, I decided to use my TRS-80 to keep track of things to be done.

"Jobjar will run on any TRS-80 computer (Modifications will be required for 4K Level I Model III's) that has an eighty column line printer.

"The DATA statements break down as follows: A description of the job to be done, the priority, the difficulty, the time

(Text continued on page 16)

```

1000 /
1010 / JOB JAR
1020 / WRITTEN BY JOHN F. STRAZZARINO
      09/80
1030 / READ AND VALIDATE
      / THE INPUT DATA
      /
1040 H=25
1050 DIM A$(H), D(H), E(H), F(H),
      K(H), L(H)
1060 FOR Q=1 TO H
1070 READ A$(Q), B$, C$, D(Q), E(Q),
      F(Q)
1080 IF A$(Q)= "END" THEN 1310
1090 IF B$= "H" THEN K(Q)=3:
      GOTO 1190
1100 IF B$= "M" THEN K(Q)=2:
      GOTO 1190
1110 IF B$= "L" THEN K(Q)=1:
      GOTO 1190
1120 PRINT A$(Q); " HAS A VALUE OF ";
      B$; " FOR PRIORITY"
1130 PRINT "THE ACCEPTABLE VALUES
      ARE:"
1140 PRINT "H FOR HIGH"
1150 PRINT "M FOR MEDIUM"
1160 PRINT "L FOR LOW"
1170 PRINT: PRINT"PLEASE FIX DATA
      LINE AND RE-RUN PROGRAM"
1180 GOTO 1730
1190 IF C$="H" THEN L(Q)=1: GOTO 1300
1200 IF C$="M" THEN L(Q)=2: GOTO 1300
1210 IF C$="E" THEN L(Q)=3: GOTO 1300
1220 PRINT A$(Q); " HAS A VALUE OF ";
      C$; " FOR DIFFICULTY"
1230 PRINT
1240 PRINT "ACCEPTABLE VALUES ARE:"
1250 PRINT "H FOR HARD"
1260 PRINT "M FOR MEDIUM"
1270 PRINT "E FOR EASY"
1280 PRINT: PRINT"FIX DATA STATEMENT
      AND RE-RUN PROGRAM"
1290 GOTO 1730
1300 NEXT Q
1310 G=Q-1
1320 /
      / SORT DATA BY PRIORITY AND
      DIFFICULTY
1330 FOR I=1 TO G-1
1340 FOR J=I+1 TO G
1350 IF K(I) > K(J) THEN 1400
1360 IF K(I)= K(J) AND L(I)>= L(J)
      THEN 1400

```

```

1370 T$=A$(I): U=K(I): V=L(I):
      W=D(I): X=E(I): Y=F(I)
1380 A$(I)=A$(J): K(I)=K(J):
      L(I)=L(J): D(I)=D(J): E(I)=E(J):
      F(I)=F(J)
1390 A$(J)=T$: K(J)=U: L(J)=V:
      D(J)=W: E(J)=X: F(J)=Y
1400 NEXT J
1410 NEXT I
1420 /
      / PRINT THE REPORT
1430 LPRINT " ": LPRINT " "
1440 LPRINT " JOB TO BE DONE";
1450 LPRINT TAB(29) "PRIORITY";
1460 LPRINT TAB(39) "DIFFICULTY";
1470 LPRINT TAB(50) "TIME";
1480 LPRINT TAB(58) "COST DATE"
1490 LPRINT " --- -- -- ----";
1500 LPRINT TAB(29) "-----";
1510 LPRINT TAB(39) "-----";
1520 LPRINT TAB(50) "----";
1530 LPRINT TAB(58) "---- ----"
1540 LPRINT " "
1550 FOR I=1 TO G
1560 LPRINT A$(I);
1570 LPRINT TAB(31);
1580 IF K(I)=3 THEN LPRINT "HIGH";
1590 IF K(I)=2 THEN LPRINT "MEDIUM";
1600 IF K(I)=1 THEN LPRINT "LOW";
1610 LPRINT TAB(41);
1620 IF L(I)=1 THEN LPRINT "HARD";
1630 IF L(I)=2 THEN LPRINT "MEDIUM";
1640 IF L(I)=3 THEN LPRINT "EASY";
1650 LPRINT TAB(50);
1660 LPRINT USING "*** HR"; D(I);
1670 M=INT(F(I)/100)
1680 N=F(I)- M* 100
1690 LPRINT TAB(57);
1700 LPRINT USING "###,## ##/##";
      E(I), M,N
1710 NEXT I
1720 LPRINT " ": LPRINT " ": GOTO 1740
1730 PRINT: PRINT "PROGRAM ABORTED"
1740 PRINT: PRINT "END OF PROGRAM":
      END
1750 DATA "PAINT FENCE", H, H, 5, 50,
      1281
1760 DATA "FIX RAIN GUTTERS", H, M,
      10, 25, 1281
1770 DATA "CLEAN BEHIND
      REFRIGERATOR", M, E, 1, 1, 981
1780 DATA "FIX LIVING ROOM LAMP", L,
      M, 1, 5, 981
1790 DATA "PATCH HALL RUG", M, M, 2,
      1, 981
1800 DATA "WASH GARAGE FLOOR", L, E,
      2, 1, 981
1810 DATA "REPAIR YARD STEPS", M, H,
      3, 50, 1081
1820 DATA "BUILD BACK FENCE", L, H,
      10, 75, 1281
1830 DATA "FIX LOOSE BOARD IN HALL",
      H, E, 1, 1, 981
1840 DATA "FIX IN-LAW BED FRAME", H,
      M, 1, 5, 981
1850 DATA END, 0, 0, 0, 0, 0

```

ADDRESS CHANGE

☐ Remove from List

☐ Change as shown

Please detach address
label and mail to
address shown above.

IF UNDELIVERABLE DO NOT RETURN

What Machine? (from page 4)

```
10 GOSUB 10000
20 IF TYPE>1 THEN F=32
   ELSE F=0
30 POKE 16409, 1
40 INPUT"NAME";A$
50 IF TYPE<2 THEN GOSUB
   11000
60 N$=A$: CLS
70 PRINT"HELLO, ";N$
80 PRINT"PLEASE PRESS
   ANY KEY (SHIFT-Q TO
   QUIT)"
90 POKE 16409, 0
100 A$=INKEY$: IF A$=""
   THEN 100
110 IF ASC(A$)=113-F
   THEN PRINT@ 200,
   "THANK YOU, NEXT";
   GOTO 30
120 PRINT@ 200, A$;
130 GOTO 90
10000 (INSERT
   SUBROUTINE)
11000 (INSERT
   SUBROUTINE)
```

Job Jar (from page 15)

needed in hours, the cost in dollars, and the desired completion date in the form MMY, where MM is a one or two digit month, and YY is the last two digits of the year. It is important to make sure that the last DATA statement is:

```
DATA END, 0, 0, 0, 0, 0
```

Otherwise, the program will terminate with an Out-of-Data statement. For more than 25 sets of data, change the variable H in line 1040.

"Part two of the program sorts the validated information. It is sorted on two fields: priority and difficulty. The sorted data is arranged so that the high priority and easy jobs are listed first, followed by high priority and medium, high priority and hard, medium priority and easy, etc.

"The final part of the program is the most important, taking the sorted data and printing a report.

"In order to use the program, substitute your information for that in lines 1750 through 1850. It is also a good idea to put just one set of data per program line. It is easier to catch mistakes that way. Have fun!"

Bugs, Errors and Fixes

26-4504 Model II Accounts Receivable

There have been numerous reports of COBOL error 98 (Invalid Index) occurring during posting.

In order to correct the ERROR 98 problem, a couple other problems we found, and add a couple of enhancements, all registered owners of 26-4504 are being mailed a free diskette with version 1.1 of the program. This diskette comes with instructions for transferring your existing data to the new version.

If you have not received this diskette, contact the Radio Shack store where you purchased the software. They should have your copy. If they don't, contact your nearest Radio Shack Computer Center or Computer Marketing Representative for a copy. You will need to present proof that you are the original purchaser of the Accounts Receivable Package.

26-4910 BASIC 1.2 Manual Error

Page 3/12A of the TRSDOS 2.0 BASIC 1.2 reference manual is the page which describes the NAME command.

Change ALL references to:

```
RENAME OLDNAME TO NEWNAME
to:
NAME OLDNAME AS NEWNAME
```

Please note that there were two distinct errors here. The first is the use of the connective " TO " instead of the proper connective " AS ." The second was the use of RENAME instead of NAME.

COMPUTER SERVICES ADDRESS AND PHONE NUMBERS

8AM to 7PM Central Time

Computer Services
900 Two Tandy Center
Fort Worth, Texas 76102

1-800-433-1679 (WATS Except Texas)
1-800-772-5914 (WATS Inside Texas)
1-817-390-3583 (Switchboard)

TRS-80 Microcomputer News: © 1981
Tandy Corporation, Fort Worth, Texas
76102 U.S.A. All Rights Reserved

Reproduction or use, without express written permission from Tandy Corporation of any portion of this Newsletter is prohibited. Permission is specifically granted to individuals to use or reproduce this material for their personal, non-commercial use. Reprint permission, with notice of source, is also specifically granted to non-profit clubs, organizations, education institutions, and Newsletters.

TRS-80 Microcomputer News is published monthly by Radio Shack, a division of Tandy Corporation. A single one year subscription is available free to purchasers of new TRS-80 Microcomputer systems with addresses in the United States, Puerto Rico, and APO or FPO addresses. Subscriptions to other addresses are not available. The subscription rate for renewals and other interested persons is twelve (\$12.00) per year, check or money order in U.S. funds. All correspondence related to subscriptions should be sent to: Microcomputer News, P.O. Box 2910, Fort Worth, Texas 76101.

Back issues of Microcomputer News are not available.

The TRS-80 Newsletter welcomes the receipt of computer programs, or other material which you would like to make available to users of TRS-80 Microcomputer systems. In order for us to reprint your submission, you must specifically request that your material be considered for reprinting in the newsletter and provide no notice that you retain copyrights or other exclusive rights in the material. This assures that our readers may be permitted to recopy and use your material without creating any legal hassles.